Claims

What is claimed is:

1. A method for providing reliable interoperation of a plurality of independent software modules, the method comprises the steps of:

a) determining selection of an object by one of the plurality of independent software modules to produce a new object selection;

b) determining object selection status, which indicates a selected object and an associated one of the plurality of independent software modules; and

c) updating the object selection status when a difference occurs between at least one of: the one of the plurality of independent software modules and the associated one of the plurality of independent software modules and the new object selection and the selected object.

2. The method of claim 1 further comprises:

evoking a function by an independent software module of the plurality of independent software modules; and

performing, by the independent software module, the function upon the selected object identified in the object selection status.

3. The method of claim 2 further comprises obtaining, by the independent software module, the function from the associated one of the plurality of independent software modules.

4. The method of claim 2 further comprises informing, by the independent software module, the associated one of the plurality of independent software modules to perform at least one of: the function upon the selected object and a corresponding function upon the selected object.

5. The method of claim 1, wherein step (c) further comprises updating the object selection status with at least one of: the new object selection and the one of the plurality of independent software modules.

6. The method of claim 1 further comprises providing a deselect message to the associated one of the plurality of independent software modules when the status is updated and when the associated one of the plurality of independent software modules is not a controlling software module.

7. The method of claim 1, wherein step (a) further comprises receiving a select message from the one of the plurality of independent software modules when the one of the plurality of independent software modules is not a controlling software module.

8. The method of claim 1 further comprises providing a select message to the associated one of the plurality of independent software modules when the status is updated and when the associated one of the plurality of independent software modules is not a controlling software module.

9. The method of claim 1, wherein the plurality of independent software modules includes at least one network management software module and a plurality of element network software modules, wherein the at least one network software module performs network level operations for a communication network and wherein each of the plurality of element network software modules performs area level operations for a corresponding portion of the communication network.

10. The method of claim 1, wherein the plurality of independent software modules includes computer applications, wherein at least some of the computer applications control independent user interfaces.

11. The method of claim 1, wherein the object includes at least one object element.

12. A method for providing reliable interoperation of a plurality of independent software modules, the method comprises the steps of:

a) maintaining object selection status that includes identity of a selected object and identity of an associated one of the plurality of independent software modules, wherein the associated one of the plurality of independent software modules selected the selected object;

b) detecting selection of a function by an independent software module of the plurality of independent software modules; and

c) performing, by one of the plurality of independent software modules, the function upon the selected object.

13. The method of claim 12, wherein step (b) further comprises detecting selection via a user interface associated with the independent software module.

14. The method of claim 12, wherein step (b) further comprises receiving notification of the function from another one of the plurality of independent software modules, wherein the another one of the plurality of independent software modules detected selection of the function via a user interface associated with the one of the plurality of independent software modules.

15. The method of claim 12, wherein step (a) further comprises:

determining selection of a new selected object by one of the plurality of independent software modules to produce a new object selection;

determining current object selection status; and updating the object selection status when a difference occurs between at least one of: the one of the plurality of independent software modules and the associated one of the plurality of independent software modules and the new object selection and the selected object.

16. The method of claim 15 further comprises updating the object selection status with at least one of: the new selected object and the one of the plurality of independent software modules.

17. The method of claim 15 further comprises providing a deselect message to the associated one of the plurality of independent software modules when the status is updated and when the associated one of the plurality of independent software modules is not a controlling software module.

18. The method of claim 15, wherein step (a) further comprises receiving a select message from the one of the plurality of independent software modules when the one of the plurality of independent software modules is not a controlling software module.

19. The method of claim 15 further comprises providing a select message to the associated one of the plurality of independent software modules when the status is updated and when the associated one of the plurality of independent software modules is not a controlling software module.

20. A software management system comprises:

a plurality of independent software modules;

a processing module; and

memory operably coupled to the processing module, wherein the memory includes operational instructions that cause the processing module to (a) determine selection of an object by one of the plurality of independent software modules to produce a new object selection; (b) determine object selection status, which indicates a selected object and an associated one of the plurality of independent software modules; and (c) update the object selection status when a difference occurs between at least one of: the one of the plurality of independent software modules and the associated one of the plurality of independent software modules and the new object selection and the selected object.

21. The software management system of claim 20, wherein the memory further comprises operational instructions that cause the processing module to enable:

an independent software module of the plurality of independent software modules to perform a function upon the selected object identified in the object selection status when the function is evoked by the independent software module.

22. The software management system of claim 21, wherein the memory further comprises operational instructions that cause the processing module to enable the independent software module to obtain the function from the associated one of the plurality of independent software modules.

23. The software management system of claim 21, wherein the memory further comprises operational instructions that cause the processing module to enable the

independent software module to inform the associated one of the plurality of independent software modules to perform at least one of: the function upon the selected object and a corresponding function upon the selected object.

24. The software management system of claim 20, wherein the memory further comprises operational instructions that cause the processing module to update the object selection status with at least one of: the new object selection and the one of the plurality of independent software modules.

25. The software management system of claim 20, wherein the memory further comprises operational instructions that cause the processing module to provide a deselect message to the associated one of the plurality of independent software modules when the status is updated and when the associated one of the plurality of independent software modules is not a controlling software module.

26. The software management system of claim 20, wherein the memory further comprises operational instructions that cause the processing module to determine selection of the object associated with the one of the plurality of independent software modules by receiving a select message from the one of the plurality of independent software modules when the one of the plurality of independent software modules is not a controlling software module.

27. The software management system of claim 20, wherein the memory further comprises operational instructions that cause the processing module to provide a select message to the associated one of the plurality of independent software modules when the status is updated and when the associated one of the plurality of independent software modules is not a controlling software module.

28. The software management system of claim 20, wherein the plurality of independent software modules includes at least one network management software module and a plurality of element network software modules, wherein the at least one network software module performs network level operations for a communication network and wherein each of the plurality of element network software modules performs area level operations for a corresponding portion of the communication network.

29. The software management system of claim 20, wherein the object includes at least one object element.

30. A software management system comprises:

a plurality of independent software modules;

a processing module; and

memory operably coupled to the processing module, wherein the memory includes operational instructions that cause the processing module to (a) maintain object selection status that includes identity of a selected object and identity of an associated one of the plurality of independent software modules, wherein the associated one of the plurality of independent software modules selected the selected object; (b) detect selection of a function by an independent software module of the plurality of independent software modules; and (c) performing, by one of the plurality of independent software modules, the function upon the selected object.

31. The software management system of claim 30, wherein the memory further comprises operational instructions that cause the processing module to detect selection of the function via a user interface associated with the independent software module.

32. The software management system of claim 30, wherein the memory further comprises operational instructions that cause the processing module to detect selection of the function by receiving notification of the function from another one of the plurality of independent software modules, wherein the another one of the plurality of independent software modules detected selection of the function via a user interface associated with the one of the plurality of independent software modules.

33. The software management system of claim 30, wherein the memory further comprises operational instructions that cause the processing module to maintain the object selection status by:

determining selection of a new selected object by one of the plurality of independent software modules to produce a new object selection;

determining current object selection status; and

updating the object selection status when a difference occurs between at least one of: the one of the plurality of independent software modules and the associated one of the plurality of independent software modules and the new object selection and the selected object.

34. The software management system of claim 33, wherein the memory further comprises operational instructions that cause the processing module to update the object selection status with at least one of: the new selected object and the one of the plurality of independent software modules.

35. The software management system of claim 33, wherein the memory further comprises operational instructions that cause the processing module to provide a deselect message to the associated one of the plurality of independent software modules when the status is updated and when the associated one of the plurality of independent software modules is not a controlling software module.

36. The software management system of claim 33, wherein the memory further comprises operational instructions that cause the processing module to determine the selection of the new selected object by receiving a select message from the one of the plurality of independent software modules when the one of the plurality of independent software modules is not a controlling software module.

37. The software management system of claim 33, wherein the memory further comprises operational instructions that cause the processing module to provide a select message to the associated one of the plurality of independent software modules when the status is updated and when the associated one of the plurality of independent software modules is not a controlling software module.